REMARKS

Claims 1-8 and 11-17 are pending in this application, of which claims 1, 4, 7, 8, 13, 15 and 16 are independent.

To further prosecution, claims 1, 4-8, 11, 13, 15, and 16 are amended for clarity as suggested by the Examiner.

For example, claim 1 is amended to clarify operation of the method within the data hazard detection circuit of the processor. Support for this amendments can be found in paragraphs [0021], [0022] of the specification, and are shown in FIG. 4 of the drawings.

Claims 4-6 are amended to clarify the relationship between elements of the claims. For example, claims 4, 5 and 6 are amended to clarify the use of the data hazard detection circuit in identifying data hazards in bypass data written to the register file by an execution unit when executing the instructions. We therefore believe the relationship between registers, the processing of instructions, writing of bypass data and detection of data hazards to be clear. Support for these amendments can be found in at least paragraphs [0001], [0021] and [0022] of the specification, and FIG. 4 of the drawings.

Claim 7 is amended to reflect tangibility of the data hazard detection circuit. Support for these amendments can be found in at least paragraph [0022] of the specification.

Claims 8, 11, 13, 15 and 16 are amended to clarify that the register file, the register ID file and the data hazard detection circuit are within the processor. Support for this amendment can be found in at least paragraphs [0021-22] of the specification.

No new matter is added with the foregoing amendments.

Claim Rejections 35 U.S.C. § 112

Claims 4-6 stand rejected under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential structural cooperative relationships of elements. Claims 4-6 are amended to clarify the relationship between the processing of the

instructions, the writing the bypass data and the detection of data hazards. In particular, the term data hazard detection logic has been replaced by the term data hazard detection circuit to clarify its nature within the processor. Reconsideration of claims 4-6 in view of 35 U.S.C. § 112, second paragraph, is respectfully requested.

Claim 7 is rejected under 35 U.S.C. § 112, second paragraph, as being incomplete for omitting essential elements. Claim 7 is amended to clarify the relationship between the data hazard detection circuit, the processor, the register file and the register ID file. Reconsideration of claim 7 in view of 35 U.S.C. § 112, second paragraph, is respectfully requested.

Claim Rejections 35 U.S.C. § 101

Claims 1, 4, 7, 13, 15, and 16 stand rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. In view of the aforementioned amendments to claims 1, 4, 7, 13, 15, and 16, Applicant's believe all 35 U.S.C. § 101 rejections have been addressed.

In particular, the term 'data hazard detection logic' has been replaced by 'data hazard detection circuit' to better reflect the tangible nature of data hazard detection within a processor. Other elements within these claims are amended to clarify their inter-relationships.

For example, claim 1 recites a method for aliasing stacked registers of a register file within a data hazard detection circuit of a processor to reduce dependency of the data hazard detection circuit upon size of the register file. The result of the method is that the data hazard detection circuit is less dependent upon the size of the register file of the processor.

Regarding claim 4, we contend that the relationship between the program instructions, execution unit, bypass data and the data hazard detection circuit is clear and tangible. Respectfully, a processor does not necessarily include memory for storing program instructions, since these instructions may be located within memory external to the processor.

Claim 7 is amended to use the term 'data hazard detection circuit' to clarify that it is circuitry within a processor.

As to claim 13, as suggested by the Examiner, claim 13 is amended to more clearly reflect that the data hazard detection logic is a circuit within a processor.

As to claims 15 and 16, the use of the term data hazard detection circuit is used to clarify the language, as suggested by the Examiner.

Reconsideration of claims 1, 4, 7, 13, 15, and 16 in view of 35 U.S.C. §101 is respectfully requested.

Claim Rejections 35 U.S.C. § 102

Claim 15 stands rejected under U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,884,070 granted to Panwar (hereinafter Panwar). Respectfully we disagree.

Amended claim 15 recites a method of data hazard detection within a processor of the type having a register file, a register ID file and a data hazard detection circuit, including:

a) aliasing each register ID within the data hazard detection circuit to two or more non-consecutive registers of the register file; and

b) determining data hazards by matching register IDs within the data hazard detection circuit.

Panwar, as shown in FIG. 1B, discloses aliasing of two consecutive single precision floating-point registers (E.g., f0 and f1) to one double-precision floating point register (e.g., f0), but does not disclose or suggest aliasing of to two or more non-consecutive registers of a register file, as required by step a) of claim 15. Clearly, registers f0 and f1 of Panwar are consecutive. See Panwar FIG. 1A. The even operand registers of Panwar, FIG. 1B, as pointed out by the Examiner in paragraph 35 of the pending office action, are aliased to consecutive registers; register f0 aliases to consecutive registers f0 and f1.

Further, Panwar makes no disclosure of aliasing within a data hazard detection circuit of a processor. Panwar, in col. 3 line 25 through col. 4 line 19 and col. 8 lines

17-54, as cited by the Examiner, discloses data dependencies between instructions, but does not disclose – anywhere – how those dependencies are detected within the processor. In fact, Panwar teaches away from data hazard detection by identifying single-precision instructions and expanding each of those instructions into two microinstructions to avoid data dependencies. See Panwar col. 6, lines 26-45. Further, Panwar does not disclose the use of a register ID file or of matching register IDs within the data hazard detection circuit to determine data hazards.

For at least these reasons, Panwar cannot anticipate claim 15. Reconsideration of claim 15 is respectfully requested.

Claim Rejections 35 U.S.C. § 103

Claims 8 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,627,985 granted to Fetterman (hereinafter Fetterman) in view of U.S. Patent No. 5,513,363 granted to Kumar et al. (hereinafter "Kumar"). Respectfully we disagree.

Amended claim 8 recites a method for data hazard detection within a processor of the type having a register file, a register ID file and a data hazard detection circuit, including:

a) aliasing each register identifier of a group of register identifiers of the register ID file to two or more registers of the register file, the register file formed of equally sized non-overlapping groups of consecutive registers, each of the register identifiers aliasing one corresponding register of each group of registers; and

b) determining data hazards within the register file by comparing one or more of the register identifiers within the data hazard detection circuit.

In paragraph 16 of the pending office action, the Examiner asserts that the 'aliasing table' of Fetterman renders step a) of claim 8 obvious. Respectfully we disagree. The aliasing table of Fetterman does not disclose one to many register aliasing, as required by step a), since the entries in the register alias table of Fetterman correspond one-to-one with architectural registers of the original macroinstruction.

See Fetterman col. 8 lines 35-40 and FIG. 3. The aliasing table of Fetterman has physical registers wherein each contains a pointer to speculative result data of corresponding registers; this is a one to one relationship, and does not represent aliasing.

Fetterman, in col. 8 lines 45-49, discloses use of a valid bit in the register alias table to indicate whether the speculative data has been retired to the appropriate committed state register; but Fetterman does not disclosure or suggest comparing register identifiers to determine data hazards as required by step b). The valid bit of Fetterman is not a register identifier and does not indicate data hazards.

Kumar does not disclose register identifiers. Kumar does not disclose one to many register aliasing. Kumar also does not disclose comparing register identifiers to determine data hazards. Therefore, Kumar cannot make up for the shortfall of Fetterman in rendering claim 8 obvious, since the combination does not teach the elements of claim 8.

For at least these reasons, Fetterman combined Kumar does not render claim 8 obvious. Reconsideration of claim 8 is respectfully requested.

As for claim 15, the aliasing table of Fetterman does not disclose or suggest one to many register aliasing or suggest matching register IDs to determine data hazards. Kumar, also does not disclose or suggest one to many register aliasing or suggest matching register IDs to determine data hazards.

For at least these reasons, the combination of Fetterman and Kumar also does not render claim 15 obvious. Reconsideration of claim 15 is respectfully requested.

Claims 4 and 5 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,826,055 granted to Wang et al. (hereinafter "Wang") in view of Panwar in view of Kumar. Respectfully we disagree.

Amended claim 4 recites a processor for processing program instructions, including:

a) a register file grouped into two or more non-overlapping equally sized groups of consecutive registers;

b) an execution unit having an array of pipelines for processing the instructions and for writing bypass data to the register file; and

c) a data hazard detection circuit for detecting data hazards in the bypass data without differentiation between corresponding registers of each group.

Wang does not disclose a register file grouped into two or more non-overlapping equally sized groups of consecutive registers, as required by element a) of claim 4.

Wang does not disclose or suggest writing the bypass data using an array of pipelines as required by element b). Wang does not disclose or suggest 'pipelines' for writing bypass data to the register file.

Wang does not disclose a data hazard detection circuit for detecting data hazards in bypass data. Teaching away from element c), Wang discloses "dependency checking logic ... is used for checking instructions for dependencies ... to determine whether one or more instructions must be executed before a subsequent instruction may be executed." See Wang col. 6, lines 57-67. The data dependency checker of Wang checks instructions and does not detect hazards in bypass data as required by element c).

As argued above, Panwar does not disclose or suggest aliasing of two or more non-consecutive registers of a register file. Further, Panwar does not disclose a data hazard detection circuit for detecting data hazards in the bypass data as required by element c).

Kumar also does not disclose aliasing of two or more non-consecutive registers of a register file as required by step a). Further Kumar does not disclose or suggest an execution unit with an array of pipelines for writing the bypass data. Kumar also does not disclose or suggest a data hazard detection circuit for detecting data hazards in the bypass data as required by element c).

Therefore, Panwar and Kumar do not overcome the shortfall of Wang in rendering claim 4 obvious. For at least these reasons, the combination of Wang, Panwar, and Kumar cannot render claim 4 obvious. Reconsideration of claim 4 is respectfully requested.

Claim 5 depends from claim 4 and benefits from like argument. However, this claim has additional features that patentably distinguish over the combination of Wang, Panwar and Kumar. For example, claim 5 recites a register ID file having a plurality of register identifiers, the data hazard detection circuit aliasing data hazard detection according to mapping of each register identifier to a corresponding register of each group of consecutive registers. In paragraph 22 of the pending office action, the Examiner asserts that the 'TAG' of Wang is the same as a register ID file of claim 5. Respectfully we disagree. Teaching away from claim 5, Wang discloses that the TAG generated by RRC 204 is assigned to each instruction. See Wang col. 9 lines 9-12. Clearly, the TAG of Wang does not correspond to registers.

Therefore, Wang, Panwar, and Kumar, even when combined, cannot render claim 5 obvious. Reconsideration of claim 5 is respectfully requested.

Claim 7 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S Patent No. 5,371,684 granted to Iadonato et al. (hereinafter "Iadonato") in view of U.S. Patent No. 6,598,149 granted to Clift (hereinafter "Clift") in view of U.S. Patent No. 5,416,749 granted to Lai (hereinafter "Lai"). Respectfully we disagree.

Amended claim 7 recites a data hazard detection circuit of a processor of the type having a register file and a register ID file providing row-to-row data hazard detection, the improvement wherein each register ID of the register ID file aliases row-to-row hazard detection of the register file within the data hazard detection circuit for two or more non-consecutive rows of the register file, thereby identifying data hazards, if any, and reducing dependency of the data hazard detection circuit upon size of the register file.

In paragraph 24 of the pending office action, the Examiner asserts that Iadonato discloses a register ID file. Respectfully we disagree. Iadonato discloses a register file 117 and that the address if the registers of the register file are used for dependency checking, but these register addresses are not the same as the register ID file of claim 7. The register ID file of claim 7 has a plurality of registers that are used for data hazard detection. See paragraph [0019] of the specification and FIG. 3 of the drawings. Iadonato does not disclose a register ID file. More specifically, claim 7 requires that each register ID of the register ID file aliases row-to-row hazard

detection of the register file within the data hazard detection circuit for two or more non-consecutive rows of the register file. On the other hand, the register addresses of Iadonato are not aliased since they are used for writing and reading data stored within the registers. Further, Iadonato does not disclose or suggest data hazard detection for two or more non-consecutive rows of the register file and does not disclose or suggest aliasing of register identifiers at all.

Clift discloses a register file, but Clift does not disclose a register ID file or suggest data hazard detection for one or more registers. Therefore, Clift does not overcome the shortfall of Iadonato in rendering claim 7 obvious.

The Examiner argues that it would have been obvious to combine Lai with Iadonato and Clift. Respectfully, we disagree. Lai discloses accessing data from a sequential-access memory device that has storage registers located at consecutively addressable rows in odd and even banks. Lai discloses that data may be accessed selectively from consecutive register rows with reduced access time. See Lai, Abstract. Specifically, the odd and even rows of Lai are enabled consecutively. See Lai col. 1, lines 28-34. Lai is a sequential access mechanism, which would not be appropriate for use with Iadonato or Clift. Particularly, since consecutive rows are accessed simultaneously in Lai, it would not be possible to alias row-to-row for two or more non-consecutive rows of the register file. Lai does not disclose or suggest data hazard detection for one or more registers, and therefore Lai also does not overcome the shortfall of Iadonato and Clift in rendering claim 7 obvious.

For at least this reason, the combination of Iadonato, Clift and Lai cannot render claim 7 obvious. Reconsideration of claim 7 is respectfully requested.

Claims 8, 11, 12, and 17 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Panwar in view of Kumar. Respectfully we disagree.

As argued above, Panwar makes no disclosure of aliasing within a data hazard detection circuit of a processor. Panwar does not disclose a register file formed of equally sized non-overlapping groups of consecutive registers. The aliasing of Panwar relates to consecutive registers. Panwar does not alias one corresponding register of each group of registers as required by step a) of claim 8 obvious. Further, as argued

above, Panwar does not determine data hazards within the register file by comparing register identifiers within a data hazard detection circuit as required by step b). Panwar does not disclose register identifiers.

As argued above, Kumar does not disclose register identifiers, does not disclose one to many register aliasing and does not disclose comparing register identifiers to determine data hazards.

As argued above, Panwar does not disclose how dependencies are detected. In particular, Panwar does not disclose hazard detection logic.

For at least these reasons, the combination of Panwar and Kumar cannot render claim 8 obvious. Reconsideration of claim 8 is respectfully requested.

Claims 11, 12 and 17 depend from claim 8 and benefit from like argument. However, these claims have additional features that patentably distinguish over Panwar and Kumar. For example, claim 11 recites that the step of aliasing increases the register file size without a corresponding increase in data hazard detection circuit. Panwar does not disclose increasing the register file size without a corresponding increase in the data hazard detection circuit; instead, it is apparent from Panwar that an increase in the size of the register file would increase the size of any hazard detection circuit. The immediate application, on the other hand, teaches that although data dependencies increase with an increase in register file size, the data hazard detection circuit does not increase because groups of registers in the register file are aliased to the same register identifiers. See paragraph [0024] of the specification and FIG. 4 of the drawings.

Claim 12 recites that the group of register identifiers contains 32 register identifiers. As argued above, Panwar does not disclose register identifiers and cannot render claim 12 obvious. In particular, Panwar does not disclose or suggest groups of 32 register identifiers within a register file as required by claim 12.

Claim 17 recites that each group of consecutive registers has 32 registers. Panwar discloses a register file of 32 registers; but Panwar does not disclose or suggest register groups (i.e., multiple collections of registers, in the same register file,

that are a subset of the total number of registers). Kumar discloses multiple physical register files, but does not disclose or suggest groups of 32 consecutive registers.

For at least these reasons, even when combined, Panwar and Kumar cannot render claims 11, 12 and 17 obvious. Reconsideration of claims 11, 12 and 17 is respectfully requested.

Examiner's indication of allowable subject matter is appreciated. In view of the above amendments and remarks, we solicit allowance of claims 1-8 and 11-17.

Applicants believe no fees are due in connection with this response. If any fee is due, please charge Deposit Account No. 08-2025.

Respectfully submitted,

By: _____

Curtis A. Vock, Reg. No. 38,356
LATHROP & GAGE L.C.
4845 Pearl East Circle, Suite 302
Boulder, CO 80301
Telephone: (720) 931-3011
Facsimile: (720) 931-3001